

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
Please do not report the images to the
Image Problem Mailbox.

PAT-NO: JP411251442A
DOCUMENT-IDENTIFIER: JP 11251442 A
TITLE: RECONFIGURABLE PROCESSOR DEVICE
PUBN-DATE: September 17, 1999

INVENTOR-INFORMATION:

NAME	COUNTRY
MARSHALL, ALAN DAVID	N/A
STANSFIELD, ANTHONY	N/A
VUILLEMIN, JEAN	N/A

ASSIGNEE-INFORMATION:

NAME	COUNTRY
HEWLETT PACKARD CO <HP>	N/A

APPL-NO: JP10359745

APPL-DATE: December 17, 1998

INT-CL (IPC): H01L021/82, H01L027/04 , H01L021/822

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a reconfigurable processor device which is low cost as well as high-performing.

SOLUTION: This reconfigurable device has a plurality of processing devices, a connection matrix providing mutual connection between the processing devices and a means defining the configuration of the connection matrix. Each of the processing devices has an arithmetic-logic unit configured to generate an output by executing a function of input operands. These operands are provided as input to the arithmetic-logic unit from the interconnection on the same path

in each cycle. A path is specified for the output from the first processing device to the second processing device so that a dynamic command is enabled by a means for determining a function executed by a second processing device.

COPYRIGHT: (C)1999,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-251442

(43) 公開日 平成11年(1999) 9月17日

(51) IntCl⁹

H 0 1 L 21/82
27/04
21/822

識別記号

F I

H 0 1 L 21/82
27/04

D
D

審査請求 未請求 請求項の数1 OL (全 22 頁)

(21) 出願番号 特願平10-359745

(22) 出願日 平成10年(1998)12月17日

(31) 優先権主張番号 9 7 3 1 0 2 2 0 . 5

(32) 優先日 1997年12月17日

(33) 優先権主張国 イギリス (GB)

(31) 優先権主張番号 9 8 1 1 7 7 6 . 5

(32) 優先日 1998年6月2日

(33) 優先権主張国 イギリス (GB)

(71) 出願人 398038580

ヒューレット・パカード・カンパニー
HEWLETT-PACKARD COM
PANY
アメリカ合衆国カリフォルニア州パロアル
ト ハノーバー・ストリート 3000

(72) 発明者 アラン・デイヴィッド・マーシャル
イギリス国ブリストル・ビーエス1・4ア
ールジェイ, マーシャンツ・ランディン
グ, トリン・ヒルズ (番地表示なし)

(74) 代理人 弁理士 古谷 馨 (外2名)

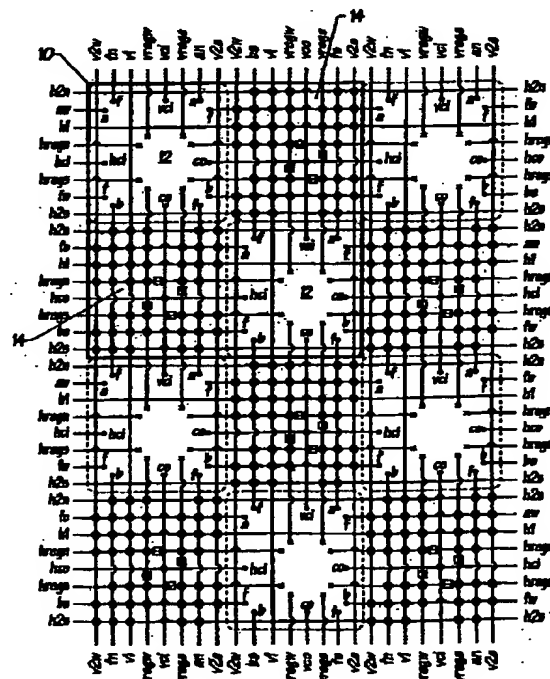
最終頁に続く

(54) 【発明の名称】 再構成可能プロセッサデバイス

(57) 【要約】

【課題】 低コストかつ高機能の再構成可能なプロセッサデバイスを提供すること。

【解決手段】 複数の処理デバイスと該処理デバイス間の相互接続を提供する接続マトリクスと該接続マトリクスの構成を規定する手段とを備えた再構成可能デバイス。前記処理デバイスの各々は入力オペランドについて関数を実行して出力を生成するよう構成された算術論理演算装置を備えている。前記入力オペランドは各サイクルで同一経路上の相互接続から前記算術論理演算装置への入力として提供される。前記処理デバイスのうちの第1の処理デバイスの出力を該処理デバイスのうちの第2の処理デバイスへと経路指定して前記第2の処理デバイスにより実行される関数を決定する手段により、動的命令がイネーブルにされる。



【特許請求の範囲】

【請求項1】複数の処理デバイスと、

該処理デバイス間の相互接続を提供する接続マトリクスと、

該接続マトリクスの構成を規定する手段と、を備えており、

前記処理デバイスの各々が、入力オペランドについて関数を実行して出力を生成するよう構成された算術論理演算装置を備えており、前記入力オペランドが、各サイクルで同一経路上の相互接続から前記算術論理演算装置への入力として提供され、前記処理デバイスのうちの第1の処理デバイスの出力を該処理デバイスのうちの第2の処理デバイスへと経路指定して前記第2の処理デバイスにより実行される関数を決定する手段が配設されていることを特徴とする、再構成可能デバイス。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、再構成可能プロセッサデバイスに関するものである。

【0002】

【従来の技術及び発明が解決しようとする課題】従来のプロセッサ（例えばIntel社のPentium II（PentiumはIntel社の商標）等）は汎用デバイスである。これは、特定のタスク用に最適化されるものではなく、非常に広範な機能を実施するようプログラムすることが可能なものである。

【0003】従来のプロセッサの汎用アーキテクチャの結果として、特定のタスクについてのプロセッサの性能は、該特定のタスクを実行するよう設計されたハードウェアよりもはるかに劣るものとなる。これは、汎用プロセッサのアーキテクチャが、タスクの構造に従っていないためではなく、タスク中にひどく酷使されると共に必然的に伴う大容量メモリリソースに対する呼び出しを極めて頻繁に行う複合ALU（算術論理演算装置）に依存するためである。かかるタスクが膨大な計算上の負荷を伴うものである場合には、汎用プロセッサを用いる方法は特に不適切なものとなる。

【0004】定期的な実行を必要とするタスクが存在する場合、適切な方法は、該タスクに特定に最適化された回路を提供することであろう。典型的な方法は、汎用プロセッサと共にコプロセッサまたはASIC（特定用途向けIC）という形でかかる回路を提供することであり、これにより、コプロセッサまたはASICが最適化されることになるタスクを汎用プロセッサにより該コプロセッサ又はASICへと経路指定することが可能となる。

【0005】ASICは特定のタスクについては最適なものであるが、特定のタスク用に構築されたものであるが故に、他の計算処理タスクについては、概して低性能のものとなり、又は全く機能しないものとなる。この2

つの極端な例、即ち、固定された構成を有するASICと（1サイクルにつきシリコンの「構成」しか存在しないものと考えられる）従来のプロセッサとの間には有利な可能性が存在する。この中間の可能性が再構成可能デバイスであり、これらは、所定の構成を有するが、必要に応じて異なる所定の構成へと再構成することが可能なものである。したがって、再構成可能デバイスは、適切な再構成により現在の計算上の必要性を満たすようそのハードウェアリソースを変更するコンピュータの可能性を提供するものとなる。

【0006】商業的に成功を収めた形の再構成可能デバイスは、フィールドプログラマブルゲートアレイ（FPGA）である。かかるデバイスは、構成可能な相互接続ネットワークに組み込まれた構成可能な処理要素の集合体からなる。相互接続構成を記述するために構成メモリが配設され、該メモリにはSRAMが使用されることが多い。これらのデバイスは、微細粒構造を有するものであり、FPGAの各処理素子は、典型的には、構成可能なゲートである。したがって、処理は、中央ALUに集中されるのではなくデバイスに分散され、該デバイスのシリコン領域がより有効に利用される。市販のFPGAシリーズの一例として、Xilinx 4000シリーズがある。

【0007】かかる再構成可能デバイスは、プロセッサ又はASICが使用されるあらゆる計算用途に概ね使用することができる。しかしながら、かかるデバイスに特に適した用途は、計算上の負荷が大きなタスクではあるが目的別に構築されるASICに値するほど一般的ではないタスクを扱うためのコプロセッサとしての用途である。したがって、再構成可能なコプロセッサは、異なるときに異なる構成でプログラムすることが可能であり、該構成の各々は、計算上の負荷の大きな異なるタスクの実行に適合するものであり、これにより、全体的なコストを著しく増大させることなく、汎用プロセッサのみの場合よりも一層優れた効率が提供される。最近のFPGAデバイスでは、その目的が動的な再構成にあり、この場合には、コードの実行中に一部又は全体の再構成が提供され、これにより、時間多重化(time-multiplexing)を用いて、1つのコードを実行する異なる段階で異なるサブタスクにとって最適化された構成を提供することが可能となる。

【0008】FPGAデバイスは、特定種類の計算タスクに特に適したものととはならない。個々の計算要素が極めて小さく、そのデータパスが極めて狭く、それらの多くが必要とされるため、構成プロセスにおいて多数の演算が必要となる。これらの構造は、小さなデータ要素について演算を行うタスクであってサイクル間で規則正しいタスクには比較的有効であるが、大きなデータ要素を伴う不規則なタスクには不十分なものとなる。かかるタスクもまた、汎用プロセッサでは良好に処理されないことが多く、（例えば画像処理等において）相当重要性の

高いものとなり得る。

【0009】代替的な再構成可能なアーキテクチャがこれまでに提案されている。その一例として、カリフォルニア大学バークレー校で開発されたPADDIアーキテクチャであり、これについては、D.Chen及びJ.Rabaeyによる「A Reconfigurable Multiprocessor IC for Rapid Prototyping of Real Time Data Paths」(ISSCC, 1992年2月)、並びに、A.Yeung及びJ.Rabaeyによる「A Data-Driven Architecture for Rapid Prototyping of High Throughput DSP Algorithms」(IEEE VLSI Signal Processing Workshop, 1992年10月)に記載されている。該アーキテクチャは、高速リアルタイムDSPシステムのプロトタイプ向けのものであり、そのDSPアルゴリズムは、従来のプロセッサ又はFPGAの何れによっても十分に扱うことができない計算処理の一例を提供するものとなる。該アーキテクチャは、再構成可能なネットワークによって接続された複数の比較的単純な処理実行装置を含むものである。その各実行装置は、16ビット幅で動作し、入力オペランド用のレジスタファイルを有し、またそれ自体の命令メモリを有している。1つの命令装置の動作を指定するために53ビットの命令ワードが必要である。

【0010】PADDIでは、構成時および実行時の両方において命令が分散される。構成時には、制御記憶手段として機能するメモリに1組の命令がロードされる。実行時には、全ての制御記憶手段に関するアドレスの大域でのブロードキャストが行われ、これらの局所命令メモリの各々は、局所実行装置により使用するためにそれ自体の局所命令を読み出す。動作時には、処理要素間の通信がデータにより駆動され、該処理要素は、それぞれの局所命令にしたがってデータに作用する。

【0011】別の代替的なアーキテクチャとしては、マサチューセッツ工科大学で開発されたMATRIXがあり、これについては、Ethan Mirsky及びAndre deHon著「MATRIX: A Reconfigurable Computing Architecture with Configurable Instruction Distribution and Deployable Resources」(FCCM '96 - IEEE Symposium on FPGAs for Custom Computing Machines, 1996年4月17~19日, Napa, California, USA)に記載されており、より詳細には、Andre deHon著「Reconfigurable Architectures for General-Purpose Computing」(第257~296頁、技術報告書1586、MIT Artificial Intelligence Laboratory)に記載されている。MATRIXは、粗粒構造を有するものであり、同一の8ビットの機能ユニットのアレイが構成可能なネットワークと相互接続されている。各機能ユニットは、256×8ビットメモリと、アドレス指定可能な入力レジスタ、出力レジスタ、及びマルチプレクサを有する8ビットALUと、制御論理回路とを備えている。該アーキテクチャは、比較的汎用性を有するものであり、即ち、FPGAの処理の非集中化を

提供すると共に、所与の用途に必要とされるものへと命令ストリームを調節するための一層幅の広いデータベースおよび有効範囲を提供するものとなる。

【0012】MATRIX構造は、有利な側面を有するものではあるが、その粗粒サイズは、従来のFPGA構造よりも多量のシリコンを消費することを意味しており、サイクル単位の規則的なタスクについては効率が悪化するものとなる。したがって、MATRIXと従来のFPGAの両方の利点を可能な限り最良に組み合わせた更なる再構成可能な構造を開発することが望ましい。

【0013】したがって、本発明は、複数の処理デバイスと、該処理デバイス間の相互接続を提供する接続マトリクスと、該接続マトリクスの構成を規定する手段とを備えた再構成可能デバイスであって、前記処理デバイスの各々が、入力オペランドについて関数を実行して出力を生成するように構成された算術論理演算装置を備えており、前記入力オペランドが、各サイクルにおける同一経路上の相互接続から前記算術論理演算装置へ入力として提供され、さらに前記複数の処理デバイスのうちの第1の処理デバイスの出力を該処理デバイスのうちの第2の処理デバイスへと経路指定して該第2の処理デバイスにより実行される関数を決定するための手段が設けられている、再構成可能デバイスを提供する。

【0014】MATRIXとは異なり、該方法は、各サイクルで同一経路上の相互接続部から入力オペランドが提供されるため、アドレス指定可能な入力レジスタを含まない(したがって入力レジスタファイルを含まない)。このため、処理パイプラインの一部として個々の処理デバイスを使用することが必要となる(命令を自分に戻すことは可能であるが、これは相互接続を介して行うことが必要になると考えられる)。したがって、MATRIXにおける個々の処理デバイスは、本発明による再構成可能デバイスにおける個別の処理デバイスよりも一層十分な範囲の関数を実施することが可能なものとなる。しかしながら、これは、所与のシリコン領域についての処理デバイスの数が増加したことにより補償されたに過ぎない。

【0015】また、本方法は、PADDIアーキテクチャに必要とされる制御記憶メモリを形成するための相当のシリコン領域の犠牲を含まないものとなる。この制御記憶メモリは、PADDIではかなりの大きさを有する必要がある、また、PADDIの実行装置は、本発明と等価の機能を得るためには本発明よりもはるかに大きなものとなる。また、制御記憶メモリは、PADDIアーキテクチャでは冗長となることが多い(各サイクル毎に同じ命令を実行するためにのみ実行装置が必要となる場合)。全ての制御記憶メモリが単一のグローバルアドレスによってアドレス指定されるPADDIの要件により、データに依存する状態でのマシンの異なる部分のシーケンス、又は計算処理の異なるスレッドについての演

算が阻止されることになる。PADDI構成では、全ての実行装置が互いに同期して実行されなければならない。

【0016】入力レジスタは、この種のアーキテクチャに必ずしも欠落したものではないことに留意されたい。即ち、アドレス指定不能な入力レジスタは本発明と両立するものである、(入力オペランドが依然として各サイクルで同一経路上で受け取られ、処理パイプラインでALUを使用する必要があるため)。しかしながら、好適な実施例では、処理デバイスは、如何なる種類の入力レジスタをも含まず、入力オペランドは、算術論理演算装置によって相互接続から直接受け取られることになる。

【0017】処理デバイスは、適当な機能を実行するための構成を必要とするものであり、動的命令を規定する何らかの基準を少なくとも設けるべきである。有利な解決策として、処理デバイスの各々が、該処理デバイスのうちの別の処理デバイスの出力により決定することが可能な第1の複数の構成ビットと、前記処理デバイスのうちの別の処理デバイスの出力により決定することが不可能な第2の複数の構成ビットと有していることが挙げられる。

【0018】

【課題を解決するための手段】好適な実施例では、各処理デバイスは、第1オペランド入力と、第2オペランド入力と、機能結果出力と、桁上げ入力と、桁上げ出力とを有しており、前記第1オペランド入力、前記第2オペランド入力、及び前記機能結果出力はnビットであり(nは1よりも大きい整数)、前記桁上げ入力及び前記桁上げ出力は1ビットである。n=4の場合に特に良好な設計上の解決策が得られる。

【0019】好適な実施例では、動的命令機構は、処理デバイスの各々が(その関数の決定のために)別の処理デバイスからのnビット命令入力を受け取るように構成される。

【0020】動的命令を提供するための更に有利な方法は、処理デバイスのうちの1つに対する前記桁上げ入力に該処理デバイスの算術論理演算装置の関数を変更することを可能にする(例えば、桁上げ入力に算術論理演算装置の関数を論理的な補数へと変更することを可能にする)手段を設けることによるものである。しかしながら、汎用性を有する演算の場合には、桁上げ入力を一定値として保持する手段を処理デバイスの各々に設けることもまた有利である。更に有利な方法として、処理デバイスのうちの第2の処理デバイスに入力された命令の2つの値間の多重化を第1の処理デバイスの桁上げ入力の値にしたがって行うために該第1の処理デバイスを使用することを可能とするものがあり、また、随意選択的に、第1の処理デバイスの桁上げ入力を該第1の処理デバイスを介して第2の処理デバイスの桁上げ入力へと伝播させるようにすることも可能である。

【0021】処理デバイスの各々が、関数出力のためのラッチ可能な出力レジスタを含むこともまた有利である。これは、「深い」パイプラインを構築する上で有用であり、この場合には、例えば、多数の演算を並列に実行し、及び異なるALUからの出力を同期させる必要がある。

【0022】個々のデバイスが動的命令を受容し又は拒絶することを可能にするために、算術論理演算装置の関数を決定する命令が相互接続から動的に受容されるべきか処理デバイスの構成メモリから提供されるべきかを決定する動的なインエーブルゲートを処理デバイスの各々に設けることが望ましい。各処理デバイスの更に有利な特徴は、動的な命令マスクであり、該動的な命令マスクを処理デバイスにより受容された命令に適用することにより、該命令を、算術論理演算装置の関数を決定する該算術論理演算装置の命令入力に供給すると共に、処理デバイスにおける周辺回路の制御のために該周辺回路の命令入力に供給することが可能となる。

【0023】

【発明の実施の形態】添付の図面を参照し、本発明の一実施例として、特定の実施例について説明することとする。

【0024】以下の説明では、「水平」、「垂直」、「北」、「南」、「東」、及び「西」という用語は、相対的な方向を理解する助けのための用いられるものであり、該用語の使用は、本発明の実施形態の絶対的な配向についての限定を示唆することを意図してはいない。

【0025】本発明の実施形態のプロセッサアレイは集積回路中に設けられる。1つのレベルでは、該プロセッサアレイは、「タイル」10からなる矩形(好適には正方形)アレイにより形成され、その内の1つが図1に太線で区切って示されている。適当な任意数のタイルを、例えば、16×16、32×32、又は64×64アレイで使うことができる。各タイル10は、矩形であり、4つの回路領域に分かれている。これらのタイルは、(接続に対称性を与えるよう)論理的に正方形であることが好ましいが、該タイルが物理的に正方形であることは重要ではない(タイミング上の対称性を提供するという点で有利な場合もあるが、一般には大して重要でない可能性が高い)。回路領域12のうちの2つは、タイル10において対角線上で対向しており、2つの算術論理演算装置(ALU)用の区域(又は場所(location))を提供するものである。タイル10において対角線上で対向する他の2つの回路領域は、一对のスウィッチングセクション14用の区域を提供するものである。

【0026】ここで図1及び図2を参照する。各ALUは、ALU内で直接接続されている第1の一对の4ビット入力aと、ALU内で直接接続されている第2の一对の4ビット入力bと、ALU内で直接接続されている4つの4ビット出力fとを有している。各ALUはまた、

独立した一対の1ビット桁上げ入力hci,vciと、ALU内で直接接続されている一対の1ビット桁上げ出力coとを有している。該ALUは、入力信号a,b,hci,vciについて標準的な演算を実行して、加算、減算、AND、NAND、OR、NOR、XOR、NXOR及び多重化等の出力信号f,coを生成することが可能であり、及び随選択的に該演算結果を登録することが可能である。個々のALUの演算については以下で一層詳細に説明することとする。ALUに対する命令は、個々の4ビットメモリセル（後述する「Hツリー」構造を介してその値を設定できるもの）から供給すること、又は後述するバスシステム上で供給することが可能である。

【0027】図1及び図2に示すレベルでは、各スイッチングセクション14は、該セクションを水平方向に横切って延びる8つのバスと、該セクションを垂直方向に横切って延びる8つのバスとを有しており、これにより、64の交差点（図2にデカルト座標で符号を付す）を有する8×8の矩形アレイが形成される。バスの全ては4ビット幅を有している（X=4における桁上げバスvc及びY=3における桁上げバスhcが1ビット幅を有することを除く）。交差点の多くには、該交差点における2つのバスを選択的に接続することが可能な4連結(gang)プログラマブルスイッチ16が設けられている。幾つかの交差点には、4連結プログラマブルスイッチ18が設けられ、該スイッチは、該交差点で端部と端部とが会合する2つのバスをそこに対して直角方向に該バスとの接続を伴うことなく選択的に接続することが可能なものである。座標(4,3)における交差点には（例えば図6(c)に示すような）プログラマブルスイッチ20が設けられており、該スイッチは、該交差点で直交する桁上げバスvc,hcを選択的に接続することが可能なものである。

【0028】次に、スイッチングセクション14における水平方向のバスについて説明する。

【0029】Y=0において、バスh2sは、プログラマブルスイッチ16により、X=0,1,2,5,6で垂直バスに接続することが可能である。該バスh2sは、2タイル分の長さを有しており、座標(4,0)ではプログラマブルスイッチ18により1つおきのスイッチングセクション14に端部同士で接続可能である。

【0030】Y=1では、ALUの入力bから西方向に延びるバスbeは、スイッチ16によりX=0,1,2,3で垂直バスに接続することが可能である。また、ALUの出力fから東方向に延びるバスfwは、スイッチ16によりX=5,6,7で垂直バスに接続することが可能である。これらのバスbe,fwの端部は、プログラマブルスイッチ18により座標(4,1)で接続することが可能である。

【0031】Y=2では、バスhregsは、プログラマブルスイッチ16によりX=1,2,3,5,6,7で垂直バスに接続することが可能である。

【0032】Y=3では、バスhcolは、ALUの桁上げ出

力coから西方向へ座標(4,3)におけるプログラマブルスイッチ20まで延びており、(a)ALUの桁上げ入力hciへと東方向に延びる桁上げバスhciに接続すること、又は(b)ALUの桁上げ入力vciへと南方向に延びる桁上げバスvciに接続することが可能である。

【0033】Y=4では、バスhregnlは、プログラマブルスイッチ16によりX=0,1,2,3,5,6で垂直バスに接続することが可能である。

【0034】Y=5では、バスhlは、X=0,1,2,3,5,6,7で垂直バスに接続することが可能である。該バスhlは、1タイル分の長さを有し、プログラマブルスイッチ18により座標(4,5)で各スイッチングセクション14において端部と端部とを接続することが可能である。

【0035】Y=6では、ALUの出力fから西方向に延びるバスfeは、スイッチ16によりX=0,1,2,3で垂直バスに接続することが可能である。また、ALUの入力aから東方向に延びるバスawは、スイッチ16によりX=5,6,7で垂直バスに接続することが可能である。バスfe,awの端部は、プログラマブルスイッチ18により座標(4,6)で接続することが可能である。

【0036】Y=7では、バスh2nlは、プログラマブルスイッチ16によりX=1,2,3,6,7で垂直バスに接続することが可能である。バスh2nlは、2タイル分の長さを有し、座標(4,7)では、プログラマブルスイッチ18により1つおきのスイッチングセクション14において端部と端部とを接続することが可能あり、座標(4,0)でバスh2sを接続するプログラマブルスイッチ18に対して互い違いに配置されている。

【0037】次に、スイッチングセクション14における垂直バスについて説明する。

【0038】X=0では、バスv2wlは、プログラマブルスイッチ16によりY=0,1,4,5,6で水平バスに接続することが可能である。該バスv2wlは、2タイル分の長さを有し、プログラマブルスイッチ18により1つおきのスイッチングセクション14において座標(0,3)で端部と端部とを接続することが可能である。

【0039】X=1では、ALUの出力fから南方向に延びるバスfnは、プログラマブルスイッチ16によりY=0,1,2で水平バスに接続することが可能である。また、ALUの入力bから北方向に延びるバスbsは、スイッチ16によりY=4,5,6,7で水平バスに接続することが可能である。バスfn,bsの端部は、プログラマブルスイッチ18により座標(1,3)で接続することが可能である。

【0040】X=2では、バスvlは、Y=0,1,2,4,5,6,7で水平バスに接続することが可能である。該バスvlは、1タイル分の長さを有し、座標(2,3)でプログラマブルスイッチ18により各スイッチングセクション14において端部と端部とを接続することが可能である。

【0041】X=3では、バスvregwlは、プログラマブルスイッチ16によりY=1,2,4,5,6,7で水平バスに接続する

10

20

30

40

50

ことが可能である。

【0042】X=4では、バスvcoは、ALUの桁上げ出力coから西方向に座標(4,3)におけるプログラマブルスイッチ20まで延びており、(a)ALUの桁上げ入力hciへと東方向に延びる桁上げバスhciに接続すること、又は(b)ALUの桁上げ入力vciへと南方向に延びる桁上げバスvciに接続することが可能である。

【0043】X=5では、バスvregelは、プログラマブルスイッチ16によりY=0,1,2,4,5,6で水平バスに接続することが可能である。

【0044】X=6では、ALUの入力aから南方向に延びるバスanは、スイッチ16によりY=0,1,2で水平バスに接続することが可能である。また、ALUの出力fから北方向に延びるバスfsは、プログラマブルスイッチ16によりX=4,5,6,7で水平バスに接続することが可能である。バスan,fsの端部は、座標(6,3)でプログラマブルスイッチ18により接続することが可能である。

【0045】X=7では、バスv2eは、プログラマブルスイッチ16によりY=1,2,5,6,7で水平バスに接続することが可能である。該バスv2eは、2タイル分の長さを有し、座標(7,3)でプログラマブルスイッチ18により1つおきのスイッチングセクション14において端部と端部を接続することが可能あり、座標(0,3)でバスv2wに接続するプログラマブルスイッチ18に対して互い違いに配置されている。

【0046】図2に示すように、バスbs,vco,fsは、スイッチングセクション14の北方向に向かってALUの入力b、出力co、及び出力fにそれぞれ接続されている。また、バスfe,hco,beは、スイッチングセクション14の西方向に向かってALUの出力f、出力co、及び入力bにそれぞれ接続されている。更に、バスaw,hci,fwは、スイッチングセクション14の東方向に向かってALUの入力a、入力ci、及び出力fにそれぞれ接続されている。更に、バスfn,vci,anは、スイッチングセクション14の南方向に向かってALUの出力f、入力ci、及び入力aにそれぞれ接続されている。

【0047】これらの接続に加えて、バスvregw,vregelは、それぞれのプログラマブルスイッチ18を介して、スイッチングセクション14の北方向に向かってALUの領域12において4ビット接続点vtsw,vtse(図2に「×」で示す)にそれぞれ接続される。また、バスhregs,hregnは、それぞれのプログラマブルスイッチ18を介して、スイッチングセクション14の西方向に向かってALUの領域12において4ビット接続点htse,htneにそれぞれ接続される。更に、バスhregs,hregnは、それぞれのプログラマブルスイッチ18を介して、スイッチングセクション14の東方向に向かってALUの領域12において4ビット接続点htsw,htnwにそれぞれ接続される。更に、バスvregw,vregelは、それぞれのプログラマブルスイッチ18を介して、スイッチングセクション14の南方向に向かって

ALUの領域12において4ビット接続点vtnw,vtneにそれぞれ接続される。これらの接続点vtnw,vtne,htne,htse,vtse,vtsw,htsw,htnwについては、図3ないし図5を参照して更に詳細に後述することとする。

【0048】また、図2に示すように、バスhregn,vreg e,hregs,vregwは、それぞれ4ビット接続点22(図2に小さな四角で示す)を有しているが、これについては、図9を参照して更に詳細を後述することとする。

【0049】図3は、算術論理演算装置の各区域の間の相互接続の1つのレベルを示すものであり、該区域の各々は、隅が丸みを帯びた四角で示されている。4つの4ビットバスv8,v4w,v4e,v16からなる1グループは、ALU区域12の各列を横切って垂直方向に延びている。各グループ中の最も左のバスv8は複数セグメントになっており、その各々は、ほぼ8タイル分の長さを有している。各グループ中の左から2つ目のバスv4wは複数セグメントになっており、その各々は、ほぼ4タイル分の長さを有している。各グループ中の右から2つ目のバスv4eは複数セグメントになっており、この場合も、その各々は、ほぼ4タイル分の長さを有しているが、左から2つ目のバスv4wとは2タイルだけオフセットされている。各グループ中の最も右のバスv16は複数セグメントになっており、その各々は、ほぼ16タイル分の長さを有している。図4の上部に位置するアレイの上縁部、及びその下縁部では、セグメントの長さは、上述したものよりも僅かに長く、又は短くすることが可能である。

【0050】ここで図3及び図5を参照する。4つのバスv8,v4w,v4e,v16からなる各グループは各ALU区域12と交差し、接続点htnw,htsw,htse,htneで4つの4ビットタップ接続が行われる。バスセグメントの端部は、ALU区域と交差するバスセグメントに対する接続上で接続されるような優先順位を取っている。

【0051】同様に、図4及び図5に示すように、4つの4ビットバスh8,h4n,h4s,h16からなるグループは、ALU区域12の各行を横切って水平方向に延びている。各グループ中の一番上のバスh8は複数セグメントになっており、その各々はほぼ8タイル分の長さを有している。各グループ中の上から2つ目のバスh4nは複数セグメントになっており、その各々はほぼ4タイル分の長さを有している。各グループ中の下から2つ目のバスh4sは複数セグメントになっており、その各々はほぼ4タイル分の長さを有しているが、上から2つ目のバスh4nから2タイル分だけオフセットされている。各グループ中の一番下のバスh16は複数セグメントになっており、その各々はほぼ16タイル分の長さを有している。図4の左側に位置するアレイの左側縁部及び右側縁部では、セグメントの長さは上記よりも僅かに長く又は短くすることが可能である。4つのバスh8,h4n,h4s,h16からなる各グループが各ALU区域12と交差する場合、更に別の4つの4ビットタップ接続が接続点vtnw,vtsw,vtse,vtneで行わ

11

れる。バスセグメントの端部は、ALU区域と交差するバスセグメントに対する接続上で接続されるような優先順位を取っている。

【0052】図5に示すように、接続点htnw,htsw,htne,htselは、プログラマブルスイッチを介して、ALU区域の西方向及び東方向に向かうスイッチングセクションのバスhreg,hregに接続されている。また、接続点vtnw,vtne,vtsw,vtseは、プログラマブルスイッチを介して、ALU区域の北方向及び南方向に向かうスイッチングセクションのバスvreg,vregに接続されている。

【0053】次に、図6(a)を参照して、直角に交差する4ビットバスからなる各対の間のプログラマブル結合部16について説明する。水平バスの導体をx0,x1,x2,x3とし、垂直バスの導体をy0,y1,y2,y3とする。ビット位置(bit significance)が同一である各導体対の間にトランジスタ160,161,162,163がそれぞれ設けられている。トランジスタ160,161,162,163のゲートはNORゲート16gの出力に共通に接続されている。該NORゲート16gは、その2つの入力に、1つのスイッチグループにより共有することが可能な単一ビットメモリセルからのイネーブル信号を反転させた信号と、単一ビットメモリセル24の内容を反転させた信号とを受信する。したがって、イネーブル信号が高レベルであり及びメモリセル24の内容が高レベルである場合にのみ、導体x0,x1,x2,x3が、トランジスタ160,161,162,163により、導体y0,y1,y2,y3にそれぞれ接続される。

【0054】次に、互いに端部と端部とが直線状に出会う4ビットバスの各対の間のプログラマブル結合部16について図6(b)を参照して説明する。一方のバスの導体をx10,x11,x12,x13とし、他方のバスの導体をx20,x21,x22,x23とする。ビット位置が同一である各導体対の間にトランジスタ180,181,182,183がそれぞれ設けられている。トランジスタ180,181,182,183のゲートは、NORゲート18gの出力に共通に接続されている。該NORゲート18gは、その2つの入力として、1つのスイッチグループにより共有することが可能な単一ビットメモリセルからのイネーブル信号を反転させた信号と、単一ビットメモリセル24の内容を反転させた信号とを受信する。したがって、イネーブル信号が高レベルであり及びメモリセル24の内容が高レベルである場合にのみ、導体x10,x11,x12,x13が、トランジスタ180,181,182,183により、導体x20,x21,x22,x23にそれぞれ接続される。

【0055】次に、桁上げ導体hco,vco,hci,vciの間のプログラマブル結合部20について、図6(c)を参照して説明する。水平方向の桁上げ出力導体hcoは、トランジスタ20hh,20hvを介して水平方向の桁上げ入力導体hci及び垂直方向の桁上げ入力導体vciにそれぞれ接続される。更に、垂直方向の桁上げ出力導体vcoは、トランジスタ20vv,20vhを介して垂直方向の桁上げ入力導体vci及び水平方向の桁上げ入力導体hciにそれぞれ接続され

12

る。トランジスタ20hh,20vvのゲートは、インバータ20iの出力に共通に接続され、トランジスタ20hv,20vhのゲート及びインバータ20iへの入力は、NORゲート20gの出力に接続されている。該NORゲート20gは、その2つの入力として、1つのスイッチグループにより共有することが可能な単一ビットメモリセルからのイネーブル信号を反転させた信号と、単一ビットメモリセル24の内容を反転させた信号とを受信する。したがって、イネーブル信号が高レベルである場合に、導体hco,vcoが、メモリセル24の内容に従って、導体hci,vciにそれぞれ接続され、又は導体vci,hciにそれぞれ接続される。

【0056】図6(a)~(c)を参照して説明した切り換え可能な結合部16,18,20がNORゲート16g,18g,20gをそれぞれ備えていることが理解されよう。図7に示すように、NORゲート16gは、典型的には4つのトランジスタ16g1,16g2,16g3,16g4により形成され、前記2つのトランジスタ16g1,16g3が反転されたイネーブル信号にตอบสนองし、前記2つのトランジスタ16g2,16g4がメモリセル24の反転された内容にตอบสนองする。切換可能な結合部16,18,20からなるグループは、かかるグループのうちの一部のみをディセーブルにする必要性を伴うことなく共通にディセーブルにすることが可能であることが望ましい。かかるグループは、1つのスイッチングセクション14中の切換可能な結合部の全て、特定のタイルにおける2つのスイッチングセクション14中の切換可能な結合部の全て、又はアレイの一層大きな領域における切換可能な結合部の全てから構成することが可能である。この場合、トランジスタ16g1は、図8に示すように、該グループ中の切換可能な結合部16,18,20の全てに共通のものとすることが可能である。これにより、ゲートに必要なトランジスタの数を25%省くことが可能となるが、図8に示すように、ゲートに連結する導体を追加する必要が生じる。

【0057】ここでは示さないが、メモリセル24の特性を用いることにより更なる簡略化が可能となる。該メモリセル24の内容と該内容の補数との両者は、出力として容易に求めることが可能なものである。したがって、図8の回路は、例えば、メモリセル24の特性を用いて実際のイネーブル信号及び反転されたイネーブル信号の両方を保持する必要性をなくすこと(メモリセルからの実際値又は補数値の何れか一方を用いて1つのイネーブル信号しか必要としないようにする等)により、配線コストを低減させることが可能になることが分かる。

【0058】図1及び図2を参照して上述したように、各スイッチングセクション14において、バスhreg,hreg,s,vreg,vregは4ビット結合部22によりレジスタ又はバッファ回路にそれぞれ接続され、次に該回路について図9を参照して更に詳細を説明する。4つの結合部22は、マルチプレクサ26の入力にそれぞれ接続されている。該マルチプレクサ26は、入力の一つを出力として選

10

20

30

40

50

択し、これをレジスタ又はバッファ28に供給する。該レジスタ又はバッファ28の出力は、4つの三状態バッファ30s, 30w, 30n, 30eに供給され、該バッファは、バスhregs, vregw, hregn, vregと結合部22に再び接続される。バッファ28が使用される場合には、バスhregs, vregw, hregn, vregのうちの選択された1つにおける4ビット信号が増幅されて、バスhregs, vregw, hregn, vregのうちの選択された別の1つに供給される。レジスタ28が使用される場合には、バスhregs, vregw, hregn, vregのうちの選択された1つにおける4ビット信号が増幅されて、

【0059】図9の構成の改善された形態を用いることにより、バス間の経路指定とは異なる目的で、バスhregs, vregw, hregn, vregのうちの選択された1つの4ビット信号を取り出すことが可能となる。マルチプレクサ26の(又は代替的な構成ではバッファ28の)適当な構造及び接続により、配線網から受け取った値をマルチプレクサ26又はバッファ28の出力として選択し(これらの選択を図9に符号260, 280で示す)、次いでその値を該スイッチボックスと関連するALUの命令を決定する際に使用することが可能となる。該構成の適用例を以下で詳述する。

【0060】この目的のためにマルチプレクサ26又はバッファ28を使用することは、ALUに命令を与えるために使用される値が、配線網を介して前方に送るために利用可能となる値でもあることを意味する。配線間で異なる値を送ることが望ましい場合には、異なるスイッチングセクション14を使用する必要がある。しかしながら、多くの構成では、ALUに送られる値により、その命令が1つの配線から別の1つの配線へと送られる値でもあると決定されることが望ましい。これは、同一の命令を多数のALUに与えること(深い処理パイプラインにおいて生じることが多い)が望ましい場合に適している。代替的な実施例(図示せず)では、マルチプレクサ26及びバッファ28からなる対が二対又は三対以上使用される。この場合、一対のマルチプレクサ/バッファ対は、関連するALUの命令入力を与える専用のものとする

ことが可能であり、一方、他のマルチプレクサ/バッファ対は、経路指定のために使用できる。

【0061】上述の構成により、アレイの周囲及び該アレイを横切る信号の経路指定に大きな柔軟性が提供される、ということが理解されよう。メモリセル24を用いたスイッチ16, 18, 20を適当にセッティングすることにより、及びマルチプレクサ26及びレジスタ又はバッファ28を適当にセッティングすることにより、バスv16, h16, v8, h8, v4e, v4w, h4n, h4sを主に使用して、アレイの縁部から特定のALUまで、ALU同士の間、及び特定のALUからアレイの縁部まで長距離にわたり信号を送ること

が可能である。これらのバスは、スイッチングセクション14により一直線に又は直角に共に結合させることが可能であり、レジスタ又はバッファ28による増幅によって伝播遅延が低減され、該レジスタ28によりパイプラインステージが導入される。また、これらのバスをその長さ方向に沿った途中の部分で引き出して(tap)、これにより特定の処理演算を実行するためのALUの設置がバスの長さにより完全に指定されることがないようにすることが可能である。したがって、信号は2つまたは3つ以上のALUに分散させることが可能である。更に、図1及び図2を参照して説明した前記の一層短い長さを有するバスを使用して、スイッチングセクション14とALUとの間で信号を経路指定することが可能であり、更に、例えば1つのALUからそれと同一の列及び行における隣接するALU又は(該バスが水平方向又は垂直方向に延びている場合であっても)対角線方向で隣接するALUまでの一層短い距離に渡って主に信号を送ることが可能である。この場合も、レジスタ又はバッファ28を使用して信号を増幅し又はプログラマブル遅延を導入することが可能である。

【0062】上記の構成では、メモリセル24は、スイッチングセクション14及びALU区域12と同程度までアレイを横切って分散される。各メモリセル24は、該メモリセル24により制御される1つ又は複数のスイッチ、マルチプレクサ、レジスタ又はバッファに隣接して配置される。これにより高い回路密度を達成することが可能になる。

【0063】次に、メモリセル24に対してデータの読み書きを行う態様、プログラマブルスイッチ16, 18, 20用のイネーブル信号をそれらのメモリセルに書き込む態様、命令及び/又は定数をALUに分散させる態様、及びクロック信号といった他の制御信号をアレイを横切って伝送する態様について説明する。これらの機能の全てについて、図10に示すように「Hツリー」構造(該構造自体は既知のものである)を用いることが可能である。ここで図10及び図11を参照する。図示の64区域の何れにもイネーブル信号を分散させるために、イネーブル信号30a及び該イネーブル信号に関する6ビットアドレス32aをデコーダ34aに供給する。該デコーダ34aは、該デコーダからの4つの分岐のうちの何れが、前記アドレスに通じ、及び該4つの分岐の全てにおけるデコーダ34bへの4ビットアドレス32bと共に、前記分岐における更なるデコーダ34bにイネーブル信号30bを供給するかを決定する。イネーブル信号30bを受け取ったデコーダ34bは、該デコーダ34bからの4つ分岐のうちの何れが、必要とされるアドレスに通じ、及び該4つの分岐の全てにおけるデコーダ34cへの4ビットアドレス32cと共に、前記分岐における更なるデコーダ34cにイネーブル信号30cを供給するかを決定する。該イネーブル信号30cを受け取ったデコーダ34cは、次いで単一ビットメモリセルにお

る格納可能な必要とされるアドレスにイネーブル信号44dを供給する。このHツリー構造の利点は、全ての宛先に対する信号経路の長さがほぼ等しいことにあり、これはクロック信号の場合に特に有利となる。

【0064】上記記載の構成の大きな利点は、例えば1つのスイッチングセクション14中、又は1つのタイル中の2つのスイッチングセクション中、又は複数のタイルからなる1つのサブアレイにおける複数のスイッチングセクション中のメモセル24のグループを、反転されたイネーブル信号により一括してディセーブルにして、これらメモセルの内容がそれらに関連するスイッチに影響を及ぼさないようにすることが可能であることにあ

る。次いで、これらのメモセル24は、アレイの配線を構成するために使用するのではなくアプリケーションにより「ユーザ」メモリとして使用することが可能となる。

【0065】次に、本発明の本実施例で用いられるALUの構造について図12を参照して説明する。図12(a)に示すように、ALUは、4つの入力A,B,I,Cinと2つの出力F,Coutとを有している。A,B,I,Fは全て4ビット幅を有するものであり、A,B,Fについて上述するように、隣接するスイッチブロックにより一般の相互接続に接続されている。Iへの入力は、図9に示されるマルチプレクサ26から抽出される。Cin及びCoutは、どちらも1ビット幅であり、上記と同様に一層限定された相互接続に接続されている。A,Bは、ALU用のオペランドを提供し、Fは出力を提供する。Cin,Coutは、桁上げ関数を提供するものではあるが、制御面でも重要なものである。IはALUの関数演算を決定する命令入力を提供する。これは、機能ユニットが1組のメモリビットにより制御されるという点で、標準的なFPGAと対照的である。この特徴の重要性及び配線網からALUまで命令入力を経路指定するために設けられた機構について以下で説明する。

【0066】ALUは、以下の4つの主要部分を有している。

【0067】4つの同一のビットスライス(bitslice)からなるALUデータバス

命令デコーダ

桁上げ/制御入力調整論理回路

スイッチブロックプログラミングインターフェイス

(これは、本発明の他の実施例ではALU自体に存在する必要のないものではあるが、ALUに該特徴が存在することによりルックアップテーブルモードでALUを使用する可能性が許容されることになる)

図12(b)は、ALUの単一ビットスライスのブロック図を示している。

【0068】2つの「入力バッファ」202,203は、経路指定網への電気的な接続を提供する手段に過ぎない。該アーキテクチャにはアドレス指定可能な入力レジスタ

(ひいてはレジスタファイル)は存在しない。ALUの機能ユニット201には各サイクルで同じ場所(配線網)からオペランドが提供される。

【0069】機能ユニット201は、2つの入力A,Bのブール関数Uを生成するルックアップテーブル(LUT)として動作する。4つの制御信号(L3,L2,L1,L0)により正確な関数がセットされ、表1に示すカルノー図が生成される。

【0070】

10 【表1】 (ALUビットスライスに関するカルノー図)

U =	B	A	0	1
	0		L ₀	L ₁
	1		L ₂	L ₃

ALUビットスライスに関するカルノー図

20 【0071】信号Liの生成について以下に説明する。

【0072】「和生成手段」204が、U及びCinのXORにより導出される和出力を提供する。

【0073】和 = U XOR Cin

Coutが、次の論理式に従って「桁上げ生成手段」205により生成される。

【0074】P = U OR L4

G = A OR L5

Cout = IF P THEN Cin ELSE G

30 ここで、Pは伝搬関数、Gは生成関数とみなすことができる。信号Liは後述の態様で生成される。

【0075】出力レジスタ206は、和出力を随意選択的にラッチする。この随意選択は、ALUプログラミングメモリの制御下で選択可能なものである。随意選択的に、同様のラッチ構成を桁上げ出力に与えることができる。これらの特徴は、同じ演算を幾つかのALUで同期させ又はタイミング制御された態様で実行する必要のある深いパイプラインでの使用に有利なものである。

40 【0076】広範な異なる考え得るビットスライスを使用することが可能である。所与のアーキテクチャにおけるビットスライスタイプの選択は、全体として該アーキテクチャが最も効率よく処理を行うことを意図した命令のタイプの関数とすることが可能である。より複雑な演算に有用な構築ブロックとして作用することができる様々な関数を使用することが可能であることは、明らかに望ましい。その他の特徴もまた望ましいものである。即ち、その1つの望ましい特徴は、幾つかのビットをその正規関数から「流用」して(divert)他の回路要素にわたる制御を可能にする能力である。もう1つの望ましい特徴は、特定の構成での動的な命令切替を必要としないA

LUの固定命令を格納する能力である。ALUをスイッチボックス（又はルックアップテーブル）用の読み出し／書き込みポートとして使用することを可能にする適当なデフォルト状態が存在することもまた望ましい。

【0077】図12(c)は、ALUの単一のビットスライスの物理的な実施態様のブロック図を示している。入力及び出力については図12(b)を参照して上述したと *

*おりである。該ビットスライスは、6つの制御信号を生成することを必要とする。このための機構について以下に説明する。該ビットスライスにより生成される有用な一組のブール関数を表2に示す。

【0078】

【表2】（ビットスライスの関数と関連する制御入力）

L ₀	L ₁	L ₂	L ₃	L ₄	L ₅	桁上げ入力値	
						0	1
1	1	1	0	1	1	A NAND B	A AND B
0	1	1	1	1	1	A OR B	A NOR B
0	1	1	0	1	1	A XOR B	A NXOR B
0	1	0	0	1	1	A AND B	A OR B
0	0	1	0	1	0	A AND B	A OR B
1	1	0	0	1	1	NOT B	B
0	0	1	1	1	0	B	NOT B
1	0	1	0	1	0	NOT A	A
0	1	0	1	1	1	A	NOT A
0	1	1	0	0	0	ADD	
1	0	0	1	0	0	SUB	
1	0	0	1	0	1	A=B	
1	1	1	0	0	0	MATCH1	
1	1	1	0	0	0	MATCH0	

ビットスライスについての関数及びそれに関連する制御入力

【0079】入力は、以下のグループに分類される。即ち、算術演算命令（ADD,SUB）、2入力ビット単位(bitwise)命令（AND,OR,NOR,XOR,NXOR）、1入力命令（A,B,NOT A,NOT B）、比較及びテスト命令（E※ 30※QUALS,MATCH1,MATCH0）である。これらの関数の出力を以下の表3にまとめる。

【0080】

【表3】（命令の出力）

名称	和関数	C_{out} 関数
ADD	$A+B$	算術桁上げ
SUBA	$A-B$	算術桁上げ
A AND B	$= A_1 \text{ AND } B_1$	$C_{out} = C_{in}$
A OR B	$= A_1 \text{ OR } B_1$	$C_{out} = C_{in}$
A NOR B	$= \text{NOT } (A_1 \text{ OR } B_1)$	$C_{out} = C_{in}$
A XOR B	$= A_1 \text{ XOR } B_1$	$C_{out} = C_{in}$
A NXOR B	$= \text{NOT } (A_1 \text{ XOR } B_1)$	$C_{out} = C_{in}$
A AND \bar{B}	$= A_1 \text{ AND } (\text{NOT } B_1)$	$C_{out} = C_{in}$
\bar{A} AND B	$= (\text{NOT } A_1) \text{ AND } B_1$	$C_{out} = C_{in}$
\bar{A} OR B	$= (\text{NOT } A_1) \text{ OR } B_1$	$C_{out} = C_{in}$
B OR A	$= A_1 \text{ OR } (\text{NOT } B_1)$	$C_{out} = C_{in}$
A	$= A_1$	$C_{out} = C_{in}$
B	$= B_1$	$C_{out} = C_{in}$
NOT A	$= \text{NOT } A_1$	$C_{out} = C_{in}$
NOT B	$= \text{NOT } B_1$	$C_{out} = C_{in}$
$A=B$	適用不能	if $A == B$ then 0, else 1
MATCH1	適用不能	A及びBのビット式ANDに 続いてワード幅にわたるOR を実施
MATCH0	適用不能	A及びBのビット式のORに 続いてワード幅にわたるAND を実施

命令に関する出力

【0081】2の補数演算が使用され、及び演算に矛盾が生じないよう演算の桁上げが提供される。MATCH関数は俗にいうものであり、MATCH1の場合には、A及びBの両方で1となる少なくとも1つの位置が存在する場合にのみ値1が返され、MATCH0の場合には、A及びBの両方で0となる少なくとも1つの位置が存在する場合にのみ値0が返される。

【0082】ビットスライスの出力を決定するためには、6つの制御信号 L_i が必要である。しかしながら、ALUの関数を決定する動的な命令は何れも5ビット以上を含まないことが望ましく、これにより、該命令が相互接続の4ビット経路に適合するものとなり、及びALU命令をそれとは別のALUの出力として提供することが可能となる。4つの命令ビット J_i から制御信号 L_i を導出するための1つの方式は次のようになる。

【0083】

【数1】

$$\begin{aligned}
 L_0 &= J_3 \\
 L_1 &= (\bar{J}_1 \text{ OR } \bar{J}_0) \text{ AND } (J_2 \text{ NAND } J_2) \\
 L_2 &= (J_1 \text{ OR } J_0) \text{ AND } (J_2 \text{ NAND } J_2) \\
 L_3 &= \bar{J}_2 \\
 L_4 &= (J_1 \text{ OR } \bar{J}_0) \text{ AND } (J_2 \text{ NAND } J_2) \\
 L_5 &= \bar{J}_0
 \end{aligned}$$

*

*【0084】しかし、これは、 C_{in} のみによって制御されるマルチプレサが設けられていないという点で欠点を有するものである。これを実施するために、 C_{in} についての依存性を与えるよう L_0 及び L_3 の定義を変更することができる。その他の定義は変更しない。

【0085】 $L_0 = \text{IF } (C_{in} \text{ AND } L_4) \text{ THEN } J_2 \text{ ELSE } J_3$
 $L_3 = \text{IF } (C_{in} \text{ AND } L_4) \text{ THEN } J_3 \text{ ELSE } J_2$

この結果得られた命令テーブルを表4として以下に示す。

【0086】

【表4】 (命令ビット及びそれに対応する関数)

21

22

J ₃	J ₂	J ₁	J ₀	L ₀	L ₁	L ₂	L ₃	L ₄	L ₅	桁上げ入力値	
										0	1
0	0	0	0	0	1	1	0	1	1	XOR	NXOR
0	0	0	1	0	0	1	0	1	0	A AND B	A OR B
0	0	1	0	0	1	0	0	1	1	A AND B	A OR B
0	0	1	1	0	1	1	0	0	0	ADD	
0	1	0	0	C _{in}	1	1	C _{in}	1	1	A OR B	A AND B
0	1	0	1		0	1		1	0	B	A
0	1	1	0		1	0		1	1	A	B
0	1	1	1	0	1	1	1	0	0	MATCHED	
1	0	0	0	C _{in}	1	1	C _{in}	1	1	A NAND B	A NOR B
1	0	0	1		0	1		1	0	NOT A	NOT B
1	0	1	0		1	0		1	1	NOT B	NOT A
1	0	1	1	1	1	1	0	0	0	MATCHI	
1	1	0	0	1	0	0	1	0	1		
1	1	0	1	1	0	0	1	0	0		
1	1	1	0	1	0	0	1	0	1	A=B	
1	1	1	1	1	0	0	1	0	0	SUB	

命令ビット及びそれに対応する関数

【0087】次に、ALUの命令ビットの出所について説明する。本発明の構成要素は、その少なくとも1つの態様では、MATRIXアーキテクチャと共有されるものであり、機能ユニットについての命令をそれとは別の機能ユニットの出力として生成する能力である。これは、MATRIXアーキテクチャでは、比較的粗粒の(coarse)機能ユニットを備えた構造で実施され、該機能ユニットの各々は、8ビットALUと、自分自身のレジスタファイルを有する入力レジスタとを備えている。一方、本実施例では、4ビットALUを備えると共にアドレス指定可能な入力レジスタを有さない遙かに微細な機能ユニットが使用される。入力レジスタが存在しないため、機能ユニットが処理パイプラインで演算を行う必要がある。かかるパイプライン計算モデルにおいて配線網からALUに命令を入力することを可能にする回路を図16に示す。

【0088】動的命令I（アレイ中で別のALUによって生成され、又は配線網にアクセスすることが可能なメモリから随意選択的に得られた4ビット命令）を含む入力信号は、配線網に対する結合部301から受け取られる。該入力信号は、上述のようにマルチプレクサ26（図9参照）を介して得ることも可能である。利用可能な選択肢が多数存在することが望ましい場合には、これは、1つ又は2つ以上の追加のALUをマルチプレクサ構成に使用することにより達成可能である。

【0089】該アーキテクチャの他の実施例では、全ての命令を動的に与えることが可能となり、この場合には信号はALUに直接渡される。しかし、本開示の実施例では、命令を動的に与えるか又は局所的に与えるかについての選択機能が存在する。配線網から入力された信号*50

*は、動的命令イネーブルゲート304を通過する。該ゲートの機能は、動的命令ビットIを該ALUにより使用可能とすること又はこれらを使用できないようにすることであり、後者の場合には、格納されている命令ビットを代替的に用いる必要がある。これは、該ALUについての構成RAMの単一ビット303により決定される。動的命令ビットを使用すべき場合には、ゲート304は、配線網から入力されたIの値を通過させる。そうでない場合には、ゲート304の出力がゼロになり、ALUへの命令40 30 入力は、格納された命令ビットとして4ビット制御レジスタ313中に既に格納されているものとなる。かかる既格納命令ビットは、例えばHツリーネットワークを介して早期にロードしておくことが可能である。

【0090】配線網301からの動的命令ビットIとして又はレジスタ313からの既格納命令ビットとしてALUに入力を与える機構が、動的命令マスク305という形で設けられる場合には、利点を得ることができる。これは、2つのゲート、即ちORゲート311及びANDゲート312を備えたものである。各ゲートへの入力（即ち動的命令イネーブルゲート304及びレジスタ313の出力）は同一である。

【0091】ORゲート311の出力は、関連する命令ビットJ_iである。ANDゲート312の出力は、ALUの他の部分に提供するための出力K_iとして利用可能なものであり、その理由については後述することとする。入力イネーブルビット303が低レベルである場合には、出力K_iは全て低レベルとなり、ビットスライス用の命令ビットJ_iが4ビット制御レジスタ313に従う。入力イネーブルビット303が高レベルであり、制御レジスタ313ビットが低レベルである場合には、外部入力I_iが命令ビットJ_iの

関連出力に経路指定され、関連するKiが低レベルにされる。入力イネーブルビット303が高レベルであり、制御レジスタ313ビットが高レベルである場合には、外部入力Iiが関連するKiに経路指定され、命令ビットJiの関連出力が高レベルにされる。

【0092】動的命令マスク305を提供する利点は、動的命令を効果的に利用するために、他の回路をALUと同期させて制御することが望ましいことが多いことである。例えば、ある場合には、ALUを加算演算と減算演算の間でスワップさせるべき場合に、桁上げチェーンの最下位ビットに供給する定数を、減算の場合には1をセットし、加算の場合には0をセットする必要がある。動的命令マスク305を設けることにより、図17に示すように、該周辺回路を制御するために制御入力を追加する必要がなくなる。該マスク回路により、動的命令入力の特定のビットをALUに供給し、及び該動的命令入力の他のビットを周辺回路に供給することが可能になる。

【0093】図17に示す場合には、制御レジスタ313は値0011を有する。その結果として、I3,I2はJ3,J2にそれぞれ接続するが、I1,I0はJ1,J0に接続する。J1,J0は、両方とも値1に固定されている。これにより、最下位ビットを処理するためのADD_LSBコード及びSUB_LSBコードを含むADD及びSUB命令コードの拡張セットが与えられる。この関数コードセットを達成するよう結果的に得られるIに関するコードは次の通りである。

【0094】

I 入力 ADD 命令コード	0000
I 入力 ADD_LSB 命令コード	0001
I 入力 SUB 命令コード	1100
I 入力 SUB_LSB 命令コード	1111

ALU命令コードJiは、ADD及びADD_LSBの両方については同じ(0011)であるが、ADDについては、Cinは、ビットスライス用のCinとなるよう単に伝播され、一方、ADD_LSBについては、ビットスライス用のCinの値は常に0である。SUB命令の位置も同様である。即ち、SUB_LSBについては、ビットスライス用のCinの値は常に1である。

【0095】したがって、周辺回路が動的命令入力により制御される際に同時に使用することができるALU命令の組み合わせが制限される。しかし、これは、実際上の問題を生じさせるものとはならない(限られた数の場合しかカバーする必要がないため)。例えば、図17に示す場合には、ADD命令とSUB命令との間に2つの共通ビットが存在する必要がある、即ち、この場合には、ビットJ1,J0が共に値1を有する。

【0096】したがって、ALUの4ビット出力は、別のALUの動的命令入力Iとして使用することが可能である。ALUの桁上げ出力もまた、別のALUの桁上げ入力として使用することが可能であり、これは動的命令を与える際に利用できる。ALUの演算を動的に変化さ

せることが可能な3つの基本的な方法が存在する。

【0097】1. 命令ビットIを一定に維持した状態で1つの関数の2つのバージョンの間で多重化を行うためにCinを使用することができる。図13に示す例は、ORとANDとの間の多重化を示している。これは、表4の右側2列に隣接して示されている関数間での多重化を可能にする。

【0098】2. Cinが同じ状態を維持している際に命令ビットIを変更することができる。これにより、Cinについて同じ要件を有する命令テーブルの同一列における関数間(例えば、桁上げ入力Cin=0のとき命令入力1000,0000をそれぞれ有するNANDとXORの間)でスワッピングを行うことが可能になる。実際には、これは、図14に示されるように、2つの命令間で多重化を行うために第2のALUを使用することにより最も容易に達成される。2つのIの値は、第1のALUの入力A,Bとして使用され、該第1のALUは、多重化関数0110を使用してプログラムされ、その出力がI入力として第2のALUに与えられる。

【0099】3. 命令及びCinの値を変更することができる。これにより、命令テーブルの右側2列における任意の2項目間でスワッピングを行うことが可能になる。図15は該構成を示すものであり、第1のALUをマルチプレクサとして使用し、及び該マルチプレクサで選択された命令が与えられる第2のALUをそのI入力として使用する点で、図14と類似したものである。この場合、関数は、XOR(命令コード0000)及びNOR(命令コード1000)であり、必要とされるCinの値はそれぞれ0,1である。代替的な命令入力間で選択を行うために多重化ALUのCinとして用いられる信号は、ここでは、多重化ALUのCoutとして伝播され、第2のALUのCinとして使用され、必要とされるI及びCinの組み合わせが得られることになる。

【0100】既述のように、ALUが隣接スイッチブロック用のメモリインターフェイス論理回路を備えているため、ルックアップテーブル操作が可能となる。したがって、一対のALU及びスイッチブロックを4入力4出力ルックアップテーブル(LUT)にすることができ、LUTアドレスを入力Aから得ると、出力和はAの任意のブール関数とすることができる。これは、ALU命令セットでは効率的に実行することができない命令に有用なものとなる(その考え得る例としては、パリティ生成、ビット回転、及びCASE文における複雑なパターンの突き合わせがある)。LUT操作は、ALU内でモードビットにより選択可能となっており、該実施例ではI入力により選択することはできない。

【0101】ALUのLUT操作中、I入力はディセーブルにされる。随意選択的に、該LUT操作中にメモリへの書き込みを行うことが可能となり、この場合には、Bがデータ入力として使用され、Cinが書き込みイネーブ

ル信号として使用される。該LUTの出力を使用してALUの出力を通常の態様で駆動することができる。

【0102】当該プロセッサアレイにおけるプロセッサの基本モードは、所定のセットの論理演算のうちの1つを2又は3つの入力で実行することである。最も単純な場合には、プロセッサにより実行される命令は、構成メモリの4ビットに静的にプログラムされ、該命令は、連続するクロックサイクル間で変化しない。しかし、上述のように、動的命令を与えることも可能である。即ち、各プロセッサの動作を決定する命令が、構成メモリの4

ビットと通常の配線網から得た4ビット入力との論理的な組み合わせとして形成される。次いで関連するプロセッサ（又はアレイの一部）の関数をサイクル毎に変更し、これにより完全な再構成に必要なコストが節約され、提供可能な命令の数が増大し、データに依存する演算に関するコストが削減される。

【0103】プロセッサ要素に動的命令を与えることができ、及び（このアレイの場合には構成メモリをユーザプレーンメモリへと変換することにより）ユーザプレーンメモリを利用することができるため、プロセッサアレイ内にCPUを構築することが可能であることが判明した。図18は、極めて単純なCPUのビットスライスの一例を示している。図18のビットスライスは最も単純に構築可能なものではあるが、これよりも遙かに複雑なビットスライスもまた実施可能であることに留意されたい。これは、RAM401等のメモリ（例えばユーザプレーンメモリへと変換された構成メモリとすることが可能）と、プロセッサ要素の算術論理演算装置（ALU）402と、プロセッサ要素の出力レジスタ403と、一般にRAMとすることが可能なコードメモリ404（及びユーザプレーンメモリへと変換された1つ又は2つ以上のプロセッサ要素とすることが可能）とを含む。RAM401は、16ワード×4ビットの大きさを有する。このCPUは、4ビット幅のデータスライスに作用するよう適合されるものである。これは、プロセッサ要素により処理可能なビットスライスが4ビット幅を有するためである。必要とされるデータバス幅が4ビットを越える場合には、1つ又は2つ以上の追加のプロセッサ要素が必要となる。

【0104】RAM401は、CPUのためのレジスタファイルとして機能し、CPUのためのデータを保持する。該CPUに関する命令は、コードメモリ404から受け取られる。かかる命令は、2種類からなり、即ち、ALU402それ自体に関する命令と、RAM401に関するアドレス及び読み出し／書き込み制御（アドレスポートから4本の専用線を介して渡されるアドレス情報（ADDRと示す）、及び上記とは別のポートを介して渡される読み取り／書き込み制御（R/Wと表される））である。有利には、該命令情報が圧縮形式でコードメモリ404に保持され、この場合には、コードメモリ404とCPUとの間

に追加の命令デコーダが必要となる。ALU402により実行された命令が結果が得られると、該結果が出力レジスタ403に格納される。該結果は、次のサイクルで、ALU402にフィードバックし、又はRAM401に格納することができる。単一サイクルで、RAM401に対する読み出し又は書き込みが行われる。

【0105】該CPUに適した単純な命令セットは以下に示す通りである（ここで、REGはレジスタ403、RAMはRAM401、及びaddrはRAM401へのアドレスである）。

【0106】REG := RAM(addr)→ワード番号（addr）をRAM401からREGにロードする

REG := NOT RAM(addr)→RAM(addr)の反転論理をREGにロードする

REG := REG NAND RAM(addr)→REG及びRAM(addr)内の値のビット単位の論理NANDを実行し、その結果をREGに格納する

REG := REG AND NOT RAM(addr)→REG内の値のビット単位の論理NAND及びRAM(addr)内の値の反転論理を実行し、その結果をREGに格納する

REG := REG + RAM(addr)→REG及びRAM(addr)内の値の算術加算を実行し、その結果をREGに格納する

REG := REG OR RAM(addr)→REG及びRAM(addr)内の値のビット単位の論理ORを実行し、その結果をREGに格納する

RAM(addr) := REG→レジスタファイルの書き込みを可能にする

更に複雑なデータバススライスを有し特定タイプの演算について最適化されたCPUを入手可能な機能ユニットにより構築することは自明のことである。必要となるALUの数はCPUに必要な命令セットに依存し、一般に、時間的な制約により、必要とされる命令セットの実行に必要なALUの数を最小限にすることが望ましい。追加のレジスタが必要となる場合に、1つ又は2つ以上の16ワード×4ビットRAM（RAM401と同様）で該レジスタを構築することは自明のことである。1つのデータバススライス内に多数のALUが存在する場合には、必要とされる命令をそれらに協働して計算させる命令入力における命令値で各々を駆動することが必要である。

【0107】2つ以上のデータバススライスが存在する場合もある。この場合には、各スライスにおける対応するALUが、共通のコードメモリからの命令入力を共有することが有利である。該命令が共通コードメモリ中で圧縮されている場合には、共有される命令デコーダを介して、ALUの各々に解凍された命令を提供することができる。桁上げは、桁上げ連鎖を使用して命令を実行する個々のALU毎に、最下位のデータバススライスから最上位のデータバススライスまで連鎖されることにな

る。

【0108】この種のCPUの命令ストリームを生成するために利用することが可能な方法は多数存在する。最も容易なものを図18に示す。命令ストリームはメモリから読み込まれる。しかし、結果的に生じるデータ依存型の実行によりデータ値から命令を導出することもまた可能である。最も汎用性のある構成は、これらの方法の両方を組み合わせて用いるものとなろう。

【0109】本発明の実施形態を例示のためにのみ記載してきたが、本発明に従いながら多数の変形及び展開を行ってもよい。例えば、本実施形態では、4ビットALUを処理装置として用いているが、これに加えて、又はその代わりに、他の形態のALUその他の処理装置を使用することも可能である。

【0110】更に、本実施形態では、ALU及びスイッチングセクションによりアレイ全体がカバーされるような説明を行ってきたが、他の種類の構成要素を該アレイに含めることも可能である。例えば、サブアレイは、上述のようにALU及びスイッチングセクションの4×4配列のタイルから構成することが可能である。また、該アレイは、4×4アレイにおける前述のサブアレイ及びメモリ、又は4×4アレイにおける前述のサブアレイ及びRISC CPUから構成することが可能である。

【0111】上記の実施例では、各ALU区域は正方形であり、各スイッチングセクションは正方形であってALU区域と同一サイズであるが、レジスタバスvregw, vreg, hreg, hregsにおける制御可能スイッチ18が、ALU区域の正方形の輪郭に侵入することに留意されたい。ALU区域は、スイッチングセクションと同一サイズである必要はなく、特に一層小さくすることが可能であり、このため、1つ又は2つ以上のバスが、1つのスイッチングセクション14から対角線方向で隣接するスイッチングセクション14へと水平方向又は垂直方向に直接進行すること、例えばバスh2s, h2w間又はバスv2e, v2w間を走らせるようにすることが可能である。

【0112】上記の実施例では、各ALUは、2つの独立した桁上げ入力vci, hci及び接続された一対の桁上げ出力coを有している。必要であれば、2つのタイプの桁上げ、即ち、マルチビット加算演算に特に用いられる隣接するALU間の高速桁上げと、より柔軟に経路指定することが可能でありデジタル直列演算に特に用いられる低速桁上げとを処理するようALUを構成することも可能である。高速桁上げは、図面を参照して上述したものと同様に構成可能なものであり、低速桁上げは、桁上げ導体と4ビットバスの特定ビットとの間のスイッチングセクション14におけるプログラマブルスイッチを用いることが可能である。

【0113】上記の実施例では、特定のビット幅、スイッチングセクションのサイズ、及びアレイのサイズについて説明してきたが、これらの値は全て適宜変更可能な

ものであることに留意されたい。また、プログラマブルスイッチ16, 18, 20は、各スイッチングセクション14中の特定区域に設けられるものとして説明したが、必要及び要求に応じて他の区域を用いることが可能である。

【0114】上記の実施例では、アレイは二次元のものであるが、(例えば、隣接層のスイッチングセクションが互いに互い違いになった状態で上記アレイの積層体を形成することにより)三次元アレイに本発明の原理を応用することも可能である。この積層体は、2層からなるものとするのが可能であるが、好適には少なくとも3層からなり、その層数は2の累乗であることが好ましい。

【0115】上記の実施例では、メモリセル24は、ゲート16g, 18g, 20gにより、メモリセルを他の目的(これを「ユーザプレーン」と表している)に使用できるよう制御を行うスイッチから分離させることができる。しかし、イネーブル信号メモリセルは、ユーザプレーンにすることができないものである。代替的な実施例では、特定のスイッチングセクション14におけるスイッチが、該スイッチングセクション14の境界においてバス中の更なるスイッチによりアレイの残りの部分から切断可能となっており、該更なるスイッチは、ユーザプレーンにすることができない更なるメモリセルにより制御される。

【0116】その他にも多数の変形及び展開を実施することが可能である。

【0117】以下においては、本発明の種々の構成要件の組み合わせからなる例示的な実施態様を示す。

【0118】1. 複数の処理デバイスと、該処理デバイス間の相互接続を提供する接続マトリクスと、該接続マトリクスの構成を規定する手段と、を備えており、前記処理デバイスの各々が、入力オペランドについて関数を実行して出力を生成するよう構成された算術論理演算装置を備えており、前記入力オペランドが、各サイクルで同一経路上の相互接続から前記算術論理演算装置への入力として提供され、前記処理デバイスのうちの第1の処理デバイスの出力を該処理デバイスのうちの第2の処理デバイスへと経路指定して前記第2の処理デバイスにより実行される関数を決定する手段が配設されていることを特徴とする、再構成可能デバイス。

【0119】2. 前記処理デバイスの各々が、アドレス指定可能な入力レジスタを備えておらず、このため、算術論理演算装置により相互接続から入力オペランドを直接受け取る、前項1に記載の再構成可能デバイス。

【0120】3. 前記処理デバイスの各々が、該処理デバイスのうちの別の処理デバイスの出力により決定することが可能な第1の複数の構成ビットと、該処理デバイスのうちの別の処理デバイスの出力により決定することが不可能な第2の複数の構成ビットとを有している、前項1又は前項2に記載の再構成可能デバイス。

【0121】4. 前記処理デバイスの各々が、第1オ

10

20

30

40

50

ペランド入力と、第2オペランド入力と、機能結果出力と、桁上げ入力と、桁上げ出力とを有しており、前記第1オペランド入力、前記第2オペランド入力、及び前記機能結果出力が n ビットであり（ n は1より大きい整数）、前記桁上げ入力及び前記桁上げ出力が1ビットである、前項1ないし前項3のいずれか1項に記載の再構成可能デバイス。

【0122】5. 前記 $n=4$ である、前項4項に記載の再構成可能デバイス。

【0123】6. 前記処理デバイスの各々が、その関数の決定のために、該処理デバイスのうちの別の処理デバイスからの n ビット命令入力を受け取るように構成されている、前項4又は前項5に記載の再構成可能デバイス。

【0124】7. 前記処理デバイスのうちの1つへの桁上げ入力が該処理デバイスの算術論理演算装置の関数を変更することを可能にする手段が設けられている、前項4ないし前項6のいずれか1項に記載の再構成可能デバイス。

【0125】8. 前記手段が、前記桁上げ入力が算術論理演算装置の関数を論理補数に変更することを可能にするものである、前項7に記載の再構成可能デバイス。

【0126】9. 桁上げ入力を一定値に保持することを可能とする手段が前記処理デバイスの各々に設けられている、前項4ないし前項8のいずれか1項に記載の再構成可能デバイス。

【0127】10. 前記処理デバイスのうちの第1の処理デバイスが、該第1の処理デバイスの桁上げ入力の値に従って、前記処理デバイスのうちの第2の処理デバイスへの命令入力の2つの値間で多重化を行うために使用可能となっている、前項4ないし前項7のいずれか1項に記載の再構成可能デバイス。

【0128】11. 前記第1の処理デバイスの桁上げ入力を該第1の処理デバイスを介して前記第2の処理デバイスの桁上げ入力へと伝播させることが可能となっている、前項10項に記載の再構成可能デバイス。

【0129】12. 前記処理デバイスの各々が、関数出力のためのラッチ可能な出力レジスタを備えている、前項1ないし前項11のいずれか1項に記載の再構成可能デバイス。

【0130】13. 前記処理デバイスの各々が、算術論理演算装置の関数を決定するための命令が相互接続から動的に受けられるべきか前記処理デバイス中の構成メモリから提供されるべきかを決定する動的イネーブルゲートを備えている、前項1ないし前項12のいずれか1項に記載の再構成可能デバイス。

【0131】14. 前記処理デバイスの各々が、動的命令マスクを備えており、該動的命令マスクを前記処理デバイスが受け取った命令に適用することにより、前記命令を、算術論理演算装置の関数を決定するための算術論

理演算装置の命令入力と、前記処理デバイスにおける周辺回路の制御のための周辺回路の命令入力との両方に提供することが可能となっている、前項1ないし前項13のいずれか1項に記載の再構成可能デバイス。

【0132】15. 中央処理装置の算術論理演算装置を形成するために1つ又は2つ以上の処理デバイスを配設し、前記中央処理装置のレジスタファイルとして第1のメモリを配設し、前記中央処理装置に命令を与えるためのコードメモリとして第2のメモリを配設する、という各ステップを有し、前記中央処理装置の前記算術論理演算装置の命令入力が前記第2メモリから提供されることを特徴とする、前項1ないし前項14のいずれか1項に記載の再構成可能デバイスから中央処理装置を構築する方法。

【0133】16. 前記第1のメモリと前記第2のメモリとの一方又は両方が、前記再構成可能デバイスの1つ又は2つ以上の構成要素をユーザプレーンメモリに再構成することにより提供される、前項15に記載の方法。

【0134】17. 前記再構成可能デバイスの前記1つ又は2つ以上の構成要素が、構成マトリクスの構成を規定するための手段の構成要素を含む、前項16に記載の方法。

【0135】18. 前記再構成可能デバイスの前記1つ又は2つ以上の構成要素が、1つ又は2つ以上の処理デバイスを含む、前項16項又は前項17に記載の方法。

【0136】19. 前記中央処理装置が、多数の算術論理演算装置を有しており、該多数の算術論理演算装置についての命令が前記第2のメモリから提供される、前項15ないし前項18のいずれか1項に記載の方法。

【0137】20. 前記第2のメモリと、前記第1のメモリ及び前記中央処理装置の前記算術論理演算装置との両方の間に命令デコード回路が配設され、命令を圧縮形式で前記第2メモリに格納できるようになっており、また前記中央処理装置での実行前に前記命令デコード回路によりデコードされるようになっている、前項15ないし前項19のいずれか1項に記載の方法。

【0138】21. 前記多数の算術論理演算装置に共通の命令デコード回路が配設される、前項21に従属する前項20に記載の方法。

40 【図面の簡単な説明】

【図1】プロセッサアレイの一部を示す説明図であり、6つのスイッチングセクション及び6つの算術論理演算装置の区域が示されている。

【図2】図1に示す構成の一部を示す部分拡大図であり、スイッチングセクションのうちの1つ及び算術論理演算装置の区域のうちの1つが示されている。

【図3】図1に示すプロセッサアレイの一部を更に小さなスケールで示す説明図であり、算術論理演算装置の区域と、これを横切って延びる「垂直」バスとが示されている。

31

【図4】図3と類似した図であるが、算術論理演算装置の区域を横切って延びる「水平」バスが示されている。

【図5】算術論理演算装置のうちの1つの区域nにおける図2、図3、及び図4のバス間の相互接続を示す説明図である。

【図6】(a)は、互いに交差する一対の4ビットバスを接続するための、スイッチングセクション中の1つのタイプのプログラマブルスイッチを詳細に示す回路図、(b)は、端部と端部が互いに接する一対の4ビットバスを接続するための、スイッチングセクション中の別のタイプのプログラマブルスイッチを詳細に示す回路図、(c)は、桁上げビットバスを接続するための、スイッチングセクション中の別のタイプのプログラマブルスイッチを詳細に示す回路図である。

【図7】図5及び図6のプログラマブルスイッチで用いることが可能な一連のNORゲートを示す回路図である。

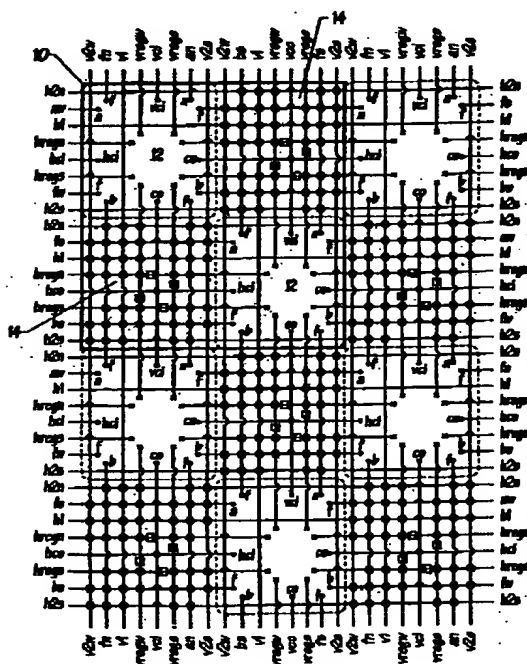
【図8】図7の回路の変形例を示す回路図である。

【図9】各スイッチングセクションで用いることが可能なバッファ及びレジスタを示す回路図である。

【図10】スイッチングセクション中のプログラマブルスイッチに対してイネーブル信号が如何に分散されるかを概略的に示す説明図である。

【図11】図10に示す構成を一層詳細に示す回路図である。

【図1】



32

【図12】(a)は、図1のアレイで用いられる個々の算術論理演算装置を示すブロック図、(b)は、個々の算術論理演算装置のビットスライスを概略的に示すブロック図、(c)は、(b)のビットスライスの物理的な実施態様を示す回路図である（これらは全て本発明の一実施例によるものである）

【図13】算術論理演算装置におけるORとANDとの間の動的な命令選択を示す説明図である。

【図14】一対の算術論理演算装置におけるNANDとXORとの間の動的な命令選択を示す説明図である。

【図15】一対の算術論理演算装置におけるXORとNORとの間の動的な命令選択を示す説明図である。

【図16】配線網から算術論理演算装置へ命令を動的に供給するための構造を示す説明図である。

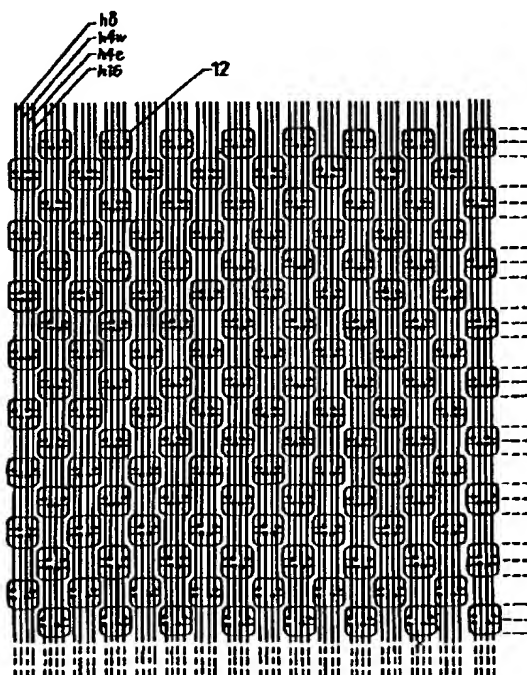
【図17】図16に示す動的命令マスクによるビットの流用(diversion)を利用した桁上げ入出力バスを示す説明図である。

【図18】極めて単純なCPUのビットスライスの一例を示す説明図である。

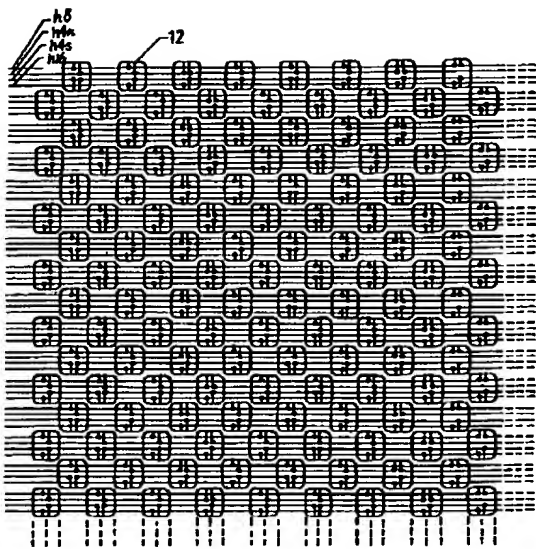
20 【符号の説明】

12	回路領域
10	タイル
14	スイッチングセクション
16, 18, 20	プログラマブルスイッチ
22	結合部

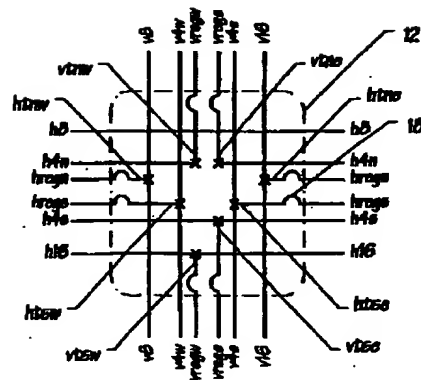
【図3】



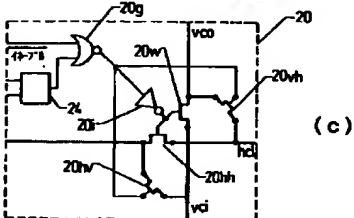
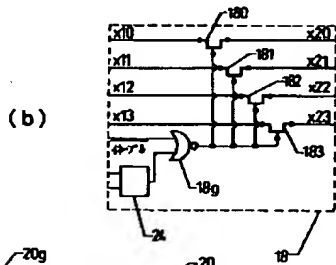
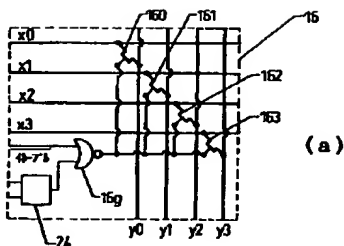
【図4】



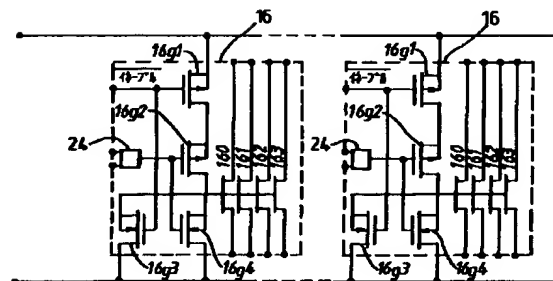
【図5】



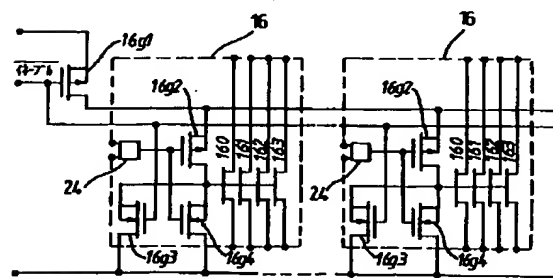
【図6】



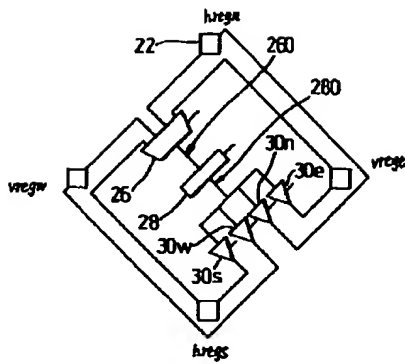
【図7】



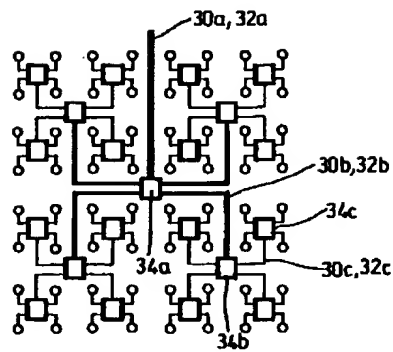
【図8】



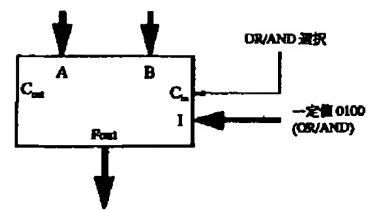
【図9】



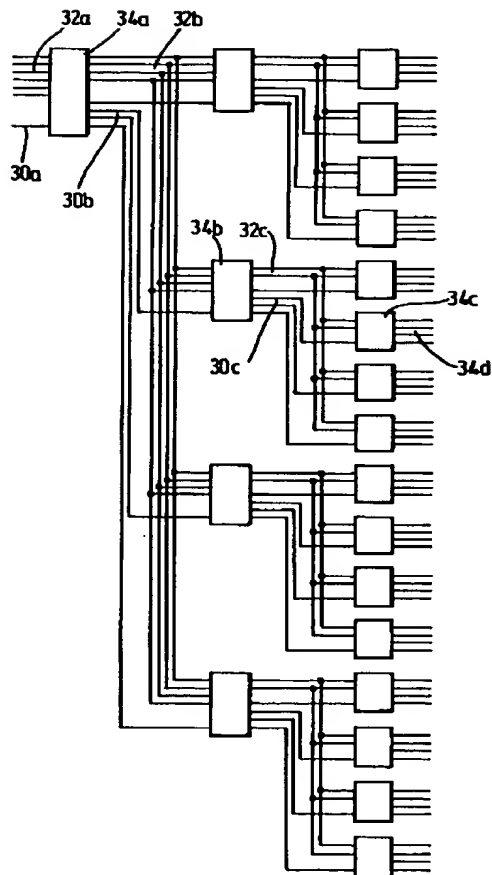
【図10】



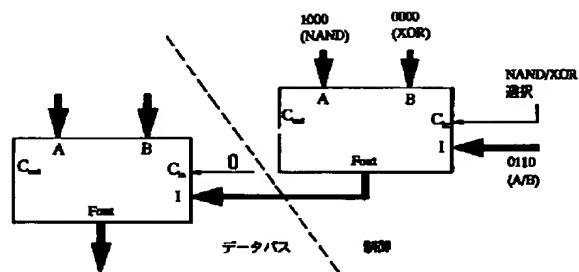
【図13】



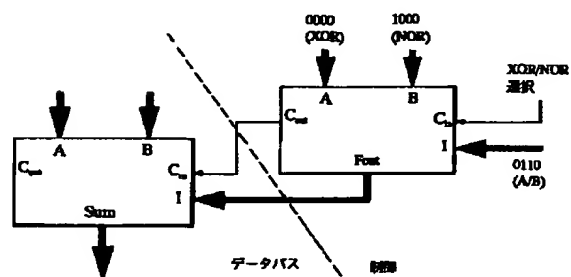
【図11】



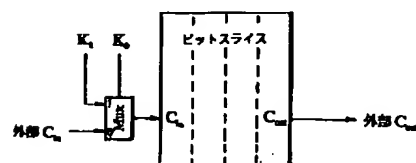
【図14】



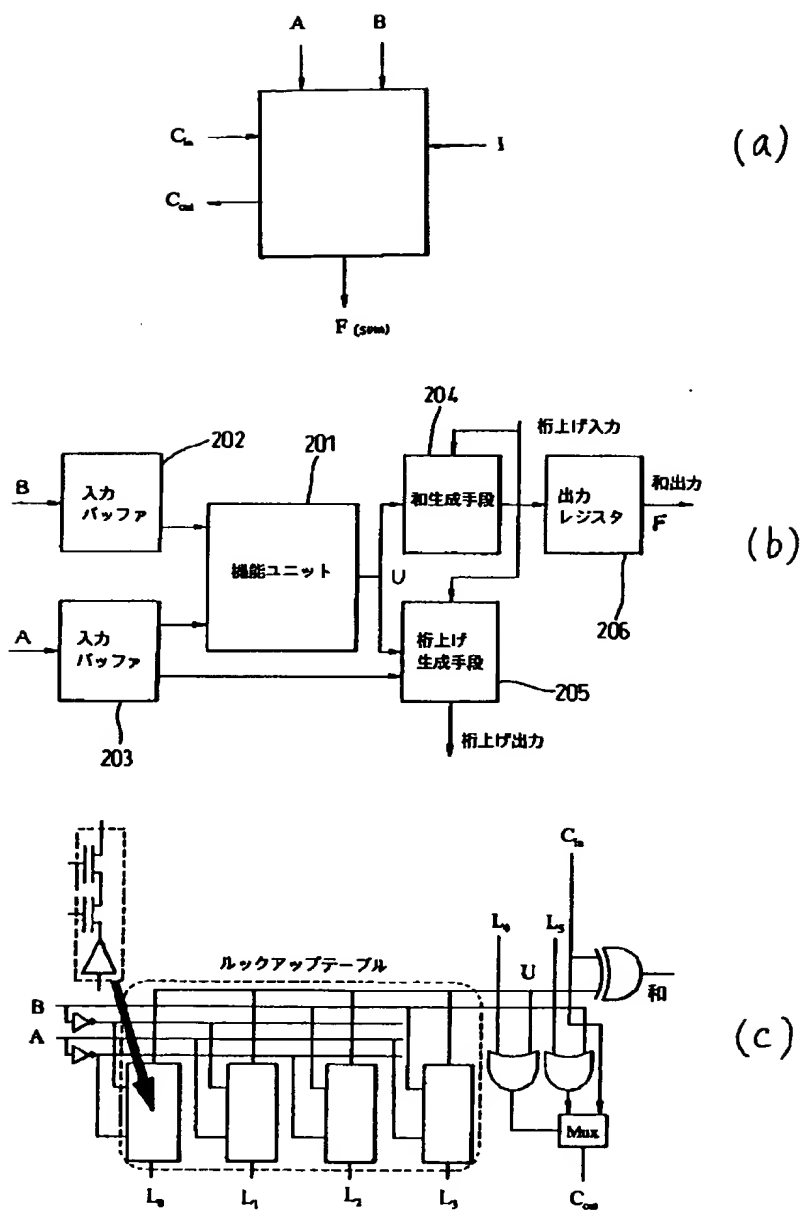
【図15】



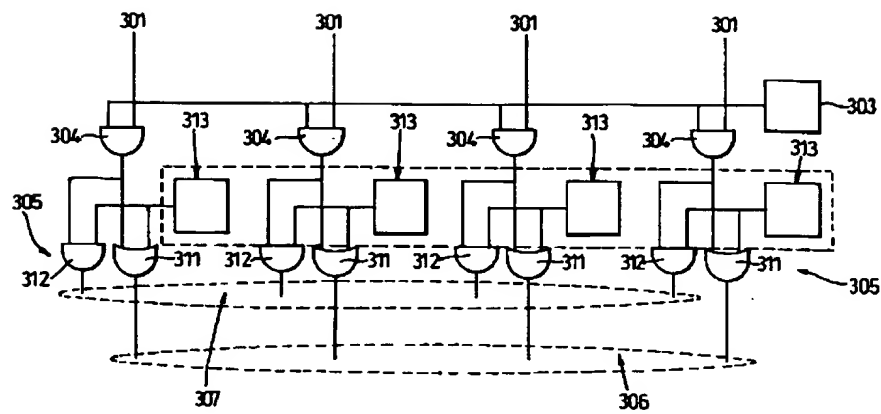
【図17】



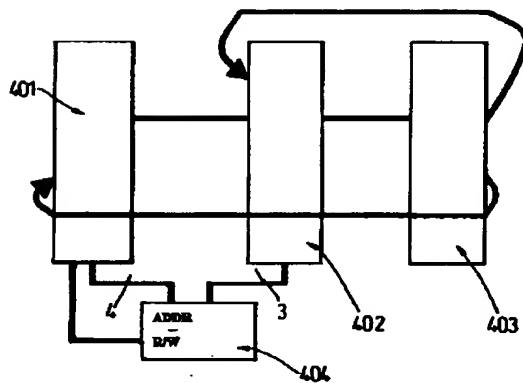
【図12】



【図16】



【図18】



フロントページの続き

(72)発明者 アンソニー・スタンスフィールド
イギリス国ブリストル・ビーエス8・4ワ
イビー, ホットウェルズ, ロウンハム・メ
ッド・60

(72)発明者 ジャン・ビュレミ
フランス国75116-パリ, リュ・ドゥ・
ラ・トゥール・76